Journal of Automated Reasoning manuscript No.
(will be inserted by the editor)

On the Fine-Structure of Regular Algebra

Simon Foster · **Georg Struth**

Received: date / Accepted: date

Abstract Regular algebra is the algebra of regular expressions as induced by regular language identity. We use Isabelle/HOL for a detailed systematic study of the regular algebra axioms given by Boffa, Conway, Kozen and Salomaa. We investigate the relationships between these systems, formalise a soundness proof for the smallest class (Salomaa's) and obtain completeness for the largest one (Boffa's) relative to a deep result by Krob. As a case study in formalised mathematics, our investigations also shed some light on the power of theorem proving technology for reasoning with algebras and their models, including proof automation and counterexample generation.

Keywords regular algebra · Kleene algebra · regular languages · interactive theorem proving · automated theorem proving · formalised mathematics

1 Introduction

Regular languages, regular expressions and finite automata belong to the foundations of computing and the canon of computer science education. Regular languages arise as the images of regular expressions under the interpretation homomorphism from regular expressions to formal languages. The kernel of this homomorphism induces a congruence on regular expressions: two expressions are equivalent if they are mapped to the same language.

Regular algebra is the algebra of regular expressions with respet to this congruence. Regular algebra axioms should be sound and complete in the following sense: an identity between two regular expressions is derivable from the these axioms if and only if these expressions are interpreted by the same regular language. The regular languages therefore form the free algebras in the variety induced by any regular algebra axioms.

Historically, research on regular algebra started with Kleene's work on the *Representation of Events in Nerve Nets and Finite Automata* [20] more than half a century ago. An

Simon Foster
Department of Computer Science, University of York, UK
E-mail: simon.foster@york.ac.uk

Georg Struth
Department of Computer Science, University of Sheffield, UK
E-mail: g.struth@sheffield.ac.uk

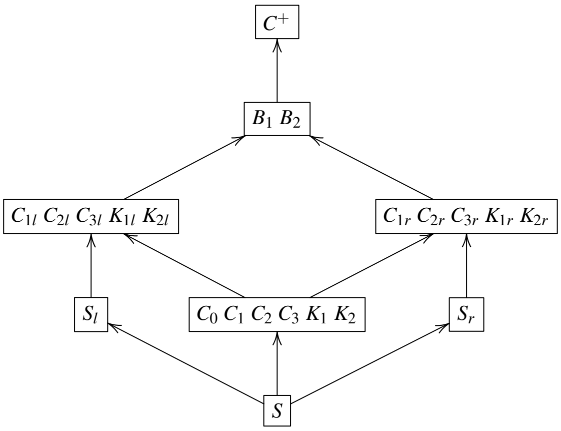
C^+
$B_1\ B_2$
$C_{1l}\ C_{2l}\ C_{3l}\ K_{1l}\ K_{2l}$
$C_{1r}\ C_{2r}\ C_{3r}\ K_{1r}\ K_{2r}$
S_l
$C_0\ C_1\ C_2\ C_3\ K_1\ K_2$
S_r
S

Fig. 1 Fine-structure of regular algebra. Arrows point at superclasses. Boxes contain equipollent classes.

early result by Redko [28] rules out any finite equational axiomatisation. Salomaa [29] gave two axiom systems based on inference rules which, essentially, encode Arden's rule from formal language theory. He proved completeness of the first one and conjectured that of the second. Conway, in his monograph on *Regular Algebras and Finite Machines* [10], conjectured completeness of several alternative axiomatisations, both with finite and infinite sets of axioms. Boffa [6] presented a particularly simple finite axiom system which is complete relative to one of Conway's conjectures; the so-called *classical axioms*—an infinite set of axioms—plus a system of monoid identities. Krob subsequently gave two long and intricate proofs of this conjecture [24] based on matrix algebras and an algebraic encoding of the Krohn-Rhodes theorem (cf. [14,11]); a deep decomposition theorem for automata and finite semigroups. His second proof has later been simplified and generalised by Ésik [12]. Relative to Boffa's algebra, Krob also verified some of Conway's remaining conjectures. Kozen proved completeness of a simplified finite axiomatisation of Conway [22], using once more encodings of automata and their classical constructions in matrix regular algebras. Independently, Bloom and Ésik [5] used the matrix technique to prove completeness of an infinite set of equational axioms, which we do not consider in this paper. Finally, Boffa [7] proved completeness of a simplified version of Kozen's axioms.

Among these variants, Kozen's axioms have—under the name *Kleene algebras*—been studied, adapted and applied most widely. Boffa's strikingly simple symmetric axioms and his elegant proofs, in contrast, have been published exclusively in French and remain virtually unknown. Most of Conway's variants have received rather limited attention as well. Thus a detailed systematic study and presentation of the fine-structure of regular algebras is certainly of interest.

Our main contribution lies in such a study. Its outcome is illustrated in Figure 1. More precisely, we consider the following axiom systems:

- weak variants of Salomaa's original axioms, called S_r in the diagram, its dual or opposite left-handed variant S_l and their combination S;
- Kozen's Kleene algebra axioms K_1 and K_2, and their left- and right-handed variants;
- Conway's systems C_0-C_3 with left- and right-handed variants of C_1-C_3;
- Boffa's systems B_1 and B_2;

- Conway's classical axioms, which are incomplete and thus not shown in the diagram; their augmentation C^{+} by an inference rule equivalent to Conway's monoid identities.

We have not attempted to formalise Bloom and Ésik's axiomatisation because it uses a matrix approach with techniques beyond the scope of this article.

An arrow in Figure 1 indicates that the algebras at its source form strict subclasses of those at its target. Axiomatisations contained in the same box are equipollent; they specify the same class. Subclass relationships are verified by proving that the subclass axioms entail the superclass ones. Strictness of entailments is verified by counterexamples. The most important one is a max-plus-style semiring constructed by Pratt [27] which we use for separating the algebras in the central diamond of Figure 1. The arrows in the diagram therefore represent strict entailment.

We also verify soundness and completeness of the axiom systems in Figure 1. For soundness we show that regular languages form models of S and hence of its superclasses. Completeness is shown relative to Krob's result for C^{+}, and thus for all subclasses.

Beyond the systems in Figure 1 we consider variants in which Kozen's and Conway's axioms are mixed and prove that all of them are either incomplete, or completeness cannot be obtained via Boffa's axioms. We also provide a variant of B_1 which axiomatises the star explicitly from a transitive closure operation and a variant of B_2 which axiomatises it explicitly as a reflexive transitive closure operation. Both are equipollent to B_1 and B_2.

Verifying axiomatic entailments can be a rather uninspiring syntactic exercise; falsifying them through counterexamples is often tedious and time consuming. Formalising the regular algebra hierarchy in the Isabelle/HOL proof assistant [25] and using its automated theorem proving and counterexample search technology [3] made these tasks pleasantly fast and automatic. Beyond pure first-order reasoning, the structures and algebras formalised include inductively defined infinite axiom sets, finitely generated algebras and set-theoretic and inductively defined models. Our Isabelle development is intended as a reference formalisation of regular algebras and their target model. Confidence in Isabelle proofs is ensured by its LCF architecture, which makes all formal developments consistent relative to a small trustworthy core. An embedding of the Isabelle/HOL axioms into ZFC set theory [19] ensures their compatibility with the most widely accepted foundations of mathematics.

Our main text presents our study of the fine-structure of regular algebra from a mathematical point of view, interspersed with comments on the Isabelle formalisation. Further formalisation details can be found in Appendix B; a brief summary of our proof experience is given in Section 13. A list of the most important axioms and axiomatisations is given in Appendix A. In the text, proofs are usually omitted. They, and the entire Isabelle development, are available online in the Archive of Formal Proofs [13].

2 Regular Algebra

This section provides the context for regular languages and regular algebra.

Let Σ be a finite alphabet and Σ^* denote the set of words over Σ, including the empty word ε. A *language* is a subset of Σ^*. We write $\mathsf{Lan}(\Sigma) = 2^{\Sigma^*}$ for the set of all languages over Σ. The *complex product* of $X, Y \in \mathsf{Lan}(\Sigma)$ is defined as the language

$$X \cdot Y = \{vw \mid v \in X \wedge w \in Y\},$$

where vw denotes the concatenation of v and w. Powers of a language X are defined inductively by $X^0 = \{\varepsilon\}$ and $X^{n+1} = X \cdot X^n$. We usually drop the multiplication symbol. The

Kleene star of a language X is defined by iteration to the first infinite ordinal as

$$X^* = \bigcup_{i\geq 0} X^i.$$

Regular languages can be obtained from regular expressions. The set of *regular expressions* over Σ is defined inductively as

$$\mathsf{Rex}(\Sigma) ::= 0 \mid 1 \mid a \in \Sigma \mid t+t \mid t\cdot t \mid t^*.$$

The operators $+$, $\cdot$ and * on regular expressions correspond to the *regular operations* on languages. Regular expressions are mapped to languages by the interpretation homomorphism $h : \mathsf{Rex}(\Sigma) \to \mathsf{Lan}(\Sigma)$ defined by

$$h(0) = \emptyset, \qquad h(1) = \{\varepsilon\}, \qquad h(a) = \{a\},$$
$$h(s+t) = h(s) \cup h(t), \qquad h(s\cdot t) = h(s)\cdot h(t), \qquad h(s^*) = h(s)^*.$$

A language is *regular* if it is the image of a regular expression under h. We write $\mathsf{Reg}(\Sigma)$ for the set of regular languages over Σ. Then $\mathsf{Reg}(\Sigma) = h(\mathsf{Rex}(\Sigma))$ and therefore

$$X \in \mathsf{Reg}(\Sigma) \Leftrightarrow \exists t \in \mathsf{Rex}(\Sigma).\ X = h(t).$$

Alternatively, the regular languages can be obtained from $\emptyset$, $\{\varepsilon\}$, and $\{a\}$, for $a \in \Sigma$, by finitely many applications of the regular operations.

The kernel of h is a congruence on regular expressions: for all $s,t \in \mathsf{Rex}(\Sigma)$,

$$s \sim t \ \Leftrightarrow\ h(s) = h(t).$$

This article studies axiom systems R that capture this congruence in the sense that

$$R \vdash s = t \ \Leftrightarrow\ s \sim t \ \Leftrightarrow\ h(s) = h(t),$$

where the same symbol is used for syntactic equality and regular language identity. General rules of first-order logic imply that $s = t$ is derivable if and only if constants from Σ are replaced by appropriate universally quantified first-order variables in $s = t$. In this sense, a universal identity in the language of regular expressions is derivable from R if and only if the terms in this identity are interpreted by the same regular language. The interpretation is obtained by replacing universally quantified variables by letters from Σ and then applying the homomorphism. Consequently, for each Σ, the regular languages over Σ form the algebras which are freely generated by Σ in the class of algebras axiomatised by R. Hence a *regular algebra* is a structure of signature $(+,\cdot,0,1,^*)$ that satisfies any such set of axioms R. As usual in mathematics we sometimes do not distinguish between axiom systems and the associated classes of algebras.

The main aim of this paper is the comparison of different regular algebra axioms. By definition, all axiomatisations have the same equational theory, but may differ in their Horn and elementary theories. For axiom systems R_1 and R_2 we write $R_1 \vdash R_2$ if every axiom in R_2 is derivable from R_1 using standard equational logic. We write $R_1 \equiv R_2$ if R_1 and R_2 are equipollent, that is, $R_1 \vdash R_2$ and $R_2 \vdash R_1$. We use the same abbreviations for axiom sets and the corresponding classes.

3 Semirings and Dioids

Dioids or idempotent semirings form a uniform basis for all regular algebras considered in this article. This deviates slightly from the concrete axiomatisations in the original articles, but it allows us to design the regular algebra hierarchy from Figure 1 in a structured modular fashion with Isabelle's type classes. For technical details on their use see Appendix B.

All regular algebras, as algebras of regular expressions, have the regular operations $(+,\cdot,0,1,^*)$ as their signature, whereas the semiring and dioid signature $(+,\cdot,0,1)$ omits the star. Historically, Salomaa [29] used dioids without 1 as a basis, since in the presence of the star, 1 can be defined as 0^*. Conway's classical axioms are based on non-idempotent semirings, but idempotency is derivable from his star axioms. Boffa follows Conway in using non-idempotent semirings in his first paper, but remarks in his second one that his axioms can be simplified in the presence of idempotency. In sum, for each of the original axiomatisations discussed in this article, there exists an equipollent dioid-based one. Hence we can freely reconstruct the fine-structure of regular algebras from the basis of dioids.

Definition 1

- A *semiring* is a structure $(S,+,\cdot,0,1)$ such that
  - $(S,+,0)$ is a commutative monoid,
  - $(S,\cdot,1)$ is a monoid,
  - the distributivity laws $x\cdot(y+z)=x\cdot y+x\cdot z$ and $(x+y)\cdot z=x\cdot z+y\cdot z$ hold for all $x,y,z\in S$,
  - the annihilation laws $0\cdot x=0$ and $x\cdot 0=0$ hold for all $x\in S$.
- A *dioid* or *idempotent semiring* is a semiring S which is additively idempotent, that is, $x+x=x$ holds for all $x\in S$.

The additive reduct $(S,+,0)$ of any dioid S forms a semilattice with least element 0 and semilattice order defined by

$$x\leq y\Leftrightarrow x+y=y.$$

In addition, the semiring operations on dioids are *isotone* or *order preserving*:

$$x\leq y\Rightarrow z+x\leq z+y,\qquad x\leq y\Rightarrow z\cdot x\leq z\cdot y,\qquad x\leq y\Rightarrow x\cdot z\leq y\cdot z.$$

Dioids are therefore ordered algebras: $(S,\leq)$ is a poset and all dioid operations preserve the ordering. Henceforth we usually write xy instead of $x\cdot y$.

An important property of semirings and dioids is opposition duality. Define the *opposite product* on a semiring S as $x\odot y=y\cdot x$. Then $S^{op}=(S,+,\odot,0,1)$ forms again a semiring; the *opposite* of S. Similarly, the opposite of a dioid is again a dioid.

Variants of semirings and dioids are available as type classes in Isabelle, including opposition duality and their most important models [1]. For our purposes only the language and regular language models are important. We use the following soundness results, which have been formalised in Isabelle.

Proposition 1 *For each alphabet Σ, the following structures form dioids.*

1. $(\mathsf{Lan}(\Sigma),\cup,\cdot,\emptyset,\{\varepsilon\})$,
2. $(\mathsf{Reg}(\Sigma),\cup,\cdot,\emptyset,\{\varepsilon\})$.

We call the first algebra the *full language dioid* over Σ. As equational classes, dioids are closed under subalgebras, products and homomorphic images by Birkhoff's theorem. Hence every subalgebra of $\mathsf{Lan}(\Sigma)$ is again a dioid, a *language dioid*. Obviously, $\mathsf{Reg}(\Sigma)$ is such a subalgebra of $\mathsf{Lan}(\Sigma)$.

The formalisation of Conway's axioms in Section 7 and of the regular language model for Salomaa's axioms in Section 11 require reasoning with finite powers and finite suprema of powers in dioids. This has not yet been formalised in Isabelle. Since the multiplicative reduct of a dioid forms a monoid and its additive reduct a commutative monoid, Isabelle's built-in functions *power* for finite powers x^i, $\in \mathbb{N}$ and *setsum* for finite sums $\sum_{i=m}^{n} x_i$ can be used to build a library for sums and powers over dioids. More abstractly, x^n is inductively defined as $x^0 = 1$ and $x^{n+1} = xx^n$. We abbreviate $x_m^n = \sum_{i=m}^{m+n} x^i$ in our formalisation.

For powers, we have built on Isabelle's extensive library which includes standard facts such as $x^n x = xx^n$ or $x^{m+n} = x^m x^n$. We have added a small number of facts which are particular to dioids, for instance $x^n \leq (x+y)^n$, $xy \leq y \Rightarrow x^n y \leq y$ and $yx \leq y \Rightarrow yx^n \leq y$.

For sums of powers we have verified a few basic facts in Isabelle, for instance $x_0^0 = 1$, $x_0^1 = 1 + x$, $x_n^0 = x^n$, $x^m + x_0^m = x_0^m$, $x \leq x_0^{n+1}$ or $1 \leq x_0^{n+1}$. Our library also contains more complex properties such as the following.

Lemma 1 *In every dioid,*

1. $x_m^{n+1} = x_m^n + x^{m+n+1}$,
2. $x_m^{i+n+1} = x_m^i + x_{m+i+1}^n$,
3. $x_0^{n+1} = 1 + x_1^n$,
4. $x_0^{m+n} + x_m^n = x_0^{m+n}$,
5. $x^k x_m^n = x_{k+m}^n$,
6. $x_0^m x_0^n = x_0^{m+n}$.

Composing and decomposing sums in Isabelle often requires user interaction; proofs about sums of powers are often inductive and therefore beyond the immediate reach of its integrated automated theorem provers. A certain amount of library design was essential for proving more complex theorems about regular algebras quickly and efficiently.

4 Conway's Classical Axioms

Conway's classical axioms can be found on p.25 of his monograph on *Regular Algebras and Finite Machines* [10]. He has shown that these axioms are incomplete—hence do not count as a regular algebra—but they are related to some of the algebras studied in this article. We therefore provide definitions.

First, we consider semirings expanded with a star operation, but without any specific star axioms. These have been called *star semirings* by Bloom and Ésik [5]. A *star dioid* is a star semiring which is also a dioid. The following definition is also due to these authors.

Definition 2

- A *Conway semiring* is a star semiring which satisfies

$$(x+y)^* = (x^* y)^* x^*, \tag{C11}$$

$$(xy)^* = 1 + x(yx)^* y. \tag{C12}$$

- A *Conway dioid* is a Conway semiring which is also a dioid.

- A *strong Conway semiring* is a Conway semiring which satisfies

$$x^{**} = x^*. \tag{C13}$$

Conway called (C11) also the *sumstar* axiom, (C12) the *prodstar* axiom and (C13) the *starstar* axiom. He observed that a Conway semiring is a Conway dioid if $1^* = 1$ holds or, equivalently, (C13). Bloom and Ésik have presented a three-element Conway dioid in which $1^* = 1$ or (C13) fails. We could automatically reproduce this counterexample with Isabelle's counterexample generator Nitpick [4], which enumerates finite models. Counterexamples presented by Nitpick can sometimes be verified internally by Isabelle. For the cases in this article, however, this was not possible due to the presence of anonymous variables in counterexample terms produced. All our counterexamples need therefore to be checked manually.

Example 1 Consider the three-element structure with addition defined by $0 < 1 < x$ and the remaining operations by the following tables.

$\cdot$	0	1	x
0	0	0	0
1	0	1	x
x	0	x	x

	$*$
0	1
1	x
x	x

In the multiplication table, only $xx = x$ is free. It can be checked that this defines a Conway dioid, but $1^* \neq x = 1$ and $0^{**} = x \neq 1 = 0^*$. Hence it is not a strong Conway dioid. □

The following fact is immediate from our previous discussion.

Proposition 2 *Every strong Conway semiring is a dioid.*

In Isabelle, we have formalised Conway dioids and strong Conway dioids (with the idempotency law being redundant), but not Conway semirings. Conway dioids form the basis of Boffa's algebras in Section 5. Strong Conway semirings are needed for some of Conway's axiomatisations in Section 7.

Finally, we can characterise algebras satisfying Conway's classical axioms.

Definition 3 A *C-algebra* is a strong Conway semiring which satisfies, for all $n \in \mathbb{N}$,

$$x^* = (x^{n+1})^* x_0^n. \tag{C14n}$$

Axiom (C14n) has been called *powerstar* axiom by Conway. Obviously, every C-algebra is again a dioid. We have therefore based C-algebras on dioids in Isabelle.

Conway has shown that the classical axioms are incomplete (p. 118, Theorem 9). There are regular expressions s and t which satisfy $s \sim t$, that is, $h(s) = h(t)$, but $C \not\vdash s = t$. His proof is based on group theory and the materialisation of his counterexample in Isabelle beyond the scope of this paper. He has also proved independence of (C13) from the other classical axioms (p.104). We have not attempted to materialise his infinite model either; Isabelle's counterexample generators Nitpick and Quickcheck [9] failed to produce a finite one.

It is easy to prove that Conway semirings or strong Conway semirings are self-duals. Showing that the dual of a C-algebra is a C-algebra in Isabelle is more involved.

Lemma 2 *In every C-algebra,*

1. $x^* = 1 + xx^*$,

2. $(xy)^*x = x(yx)^*$,
3. $x^m x^{n*} = x^{n*} x^m$, *for* $m \leq n$,
4. $x^* = x_0^n x^{(n+1)*}$.

The last identity is the dual of (C14n). This implies the following fact.

Proposition 3 *The opposite of a C-algebra is a C-algebra.*

Many additional equations can be derived in the setting of Conway semirings. Since we need to derive them in the context of Boffa's second axiomatisation, which is not based on Conway semirings, we have not considered this further. It would lead to a duplication of labour which seems of little interest for the purpose of this article.

5 Boffa's Axioms

Boffa has presented two axiom systems which are complete relative to Krob's result, as mentioned in the introduction. In the diagram of Figure 1 they are shown as B_1 and B_2 directly below Conway's infinitary axioms which are used in the relative completeness proof in Section 6. In his first article [6], Boffa has presented an axiomatisation which adds a simple symmetric quasi-identity to those of C-algebras. He has shown that a schematic rule equivalent to Conway's monoid identities can be derived from these axioms. Krob has then proved completeness for C-algebras plus this rule [24]. In his second article [7], Boffa has simplified this first axiomatisation to what we call B_1. He has also presented a second axiomatisation—which we call B_2—which entails B_1. Relative to Krob's result, this yields two strikingly simple symmetric axiomatisations of regular algebra.

This section presents Boffa's axiom systems together with a proof of equipollence. It is also shown that either of them entails Conway's classical axioms. Finally, we present two new variants of Boffa's algebras—which we call B_1' and B_{rtc}—and show that they are equipollent to both B_1 and B_2.

Definition 4

- A B_1-*algebra* is a Conway dioid which satisfies
$$xx = x \ \Rightarrow \ x^* = 1 + x. \tag{R}$$
- A B_2-*algebra* is a star dioid which satisfies
$$1 + x \leq x^*, \tag{B21}$$
$$x^* x^* = x^*, \tag{B22}$$
$$1 + x \leq y \wedge yy = y \ \Rightarrow \ x^* \leq y. \tag{B23}$$

B_1- and B_2-algebras are obviously self-dual, and we have formally verified this with Isabelle.

Proposition 4 $B_1 \equiv B_2$.

In Isabelle, B_1 and B_2 have been formalised as type classes. System B_1 expands Conway dioids whereas B_2 expands dioids. Proving that $B_1 \vdash B_2$ in Isabelle means giving an **instance** or **sublocale** proof. Isabelle then dictates the proof obligations. Since only first-order quasi-identities are involved in these proofs, they are ideally suited for Isabelle's external automated theorem provers, which are invoked by the *sledgehammer* command. The proof outputs of these external tools are internally verified by Isabelle to increase trustworthiness.

Proving $B_1 \vdash B_2$ was, in fact, fully automatic. Proving $B_2 \vdash B_1$ directly was not possible within Sledgehammer's default time limits; it required the presence of several lemmas in the context of B_2. Since B_1 and B_2 are the weakest regular algebras known with finite axiom sets, it is worthwhile to derive identities between regular expressions and other facts in this setting. They then become automatically available in all regular algebras which entail these axioms. Our Archive files contain more than 50 statements for B_2. Here we list only a few well known facts.

Lemma 3 *In every B_2-algebra,*

1. $x^*x^* = x^*$,
2. $x^{**} = x^*$,
3. $1 + xx^* = x^*$,
4. $(xy)^*x = x(yx)^*$,
5. $(x+y)^* = (x^*y)^*x^*$,
6. $(x+y)^* = (x^*y^*)$,
7. $(1+x)^* = x^*$,
8. $1 + x + x^*x^* \leq x^*$,
9. $x \leq y \Rightarrow x^* \leq y^*$.

Most of their proofs are fully automatic in Isabelle. Due to self-duality of B_2, opposite statements hold as well. Next we list some properties that involve powers. These have been verified with Isabelle by induction, using properties like those in Lemma 3.

Lemma 4 *In every B_2-algebra,*

1. $x^n \leq x^*$,
2. $x^m x^{n*} = x^{n*} x^m$, *if* $m \leq n$,
3. $x_m^n \leq x^*$,
4. $x_0^m x^{n*} = x^{n*} x_0^m$, *if* $m \leq n$,
5. $y \leq yx + z \Rightarrow y \leq yx^{n+1} + zx^*$,

Again, opposite statements hold by self-duality of B_2. Lemma 4(5), in particular, is instrumental for proving Arden's rule in the language model in Section 11.

We have also formalised Boffa's proof that Conway's classical axioms can be derived from B_1 and hence B_2.

Proposition 5 *$B_2 \vdash C$ (and $B_1 \vdash C$).*

The proof in Isabelle requires, in particular, the derivation of the powerstar axiom (C14n), which needs some preparatory lemmas. Obviously, C neither entails B_1 nor B_2, simply because Boffa's algebras are complete whereas C is not. We have unsuccessfully tried to obtain finite counterexamples to $C \vdash B_1$ with Isabelle"'s counterexample generators. Given the compexity of Conway's counterexample mentioned on p.7, this is no surprise. As a general observation, Nitpick and Quickcheck failed in the presence of the powerstar axiom.

Axiom (R) almost relates a transitive closure with a reflexive transitive closure operation as modelled by the star. In relation algebra a relation x is transitive if $xx \leq x$ and a transitive relation is equal to its transitive closure: $xx \leq x$ implies $x = x^+$. Moreover, the transitive closure and the reflexive transitive closure of a relation are related by $x^* = 1 + x^+$, hence $x^* = 1 + x$ if x is transitive. Since binary relations form B_1 algebras (it is well known that they form Kleene algebras, hence the B_1 axioms are derivable according to Section 8) it is obvious that the B_1 axioms almost capture this relationship. However, (R) uses $xx = x$ instead of $xx \leq x$ as its antecedent. We have therefore added an explicit variant of B_1.

Definition 5 A B_1'*-algebra* is a Conway dioid which satisfies

$$xx \le x \ \Rightarrow\ x^* = 1 + x. \tag{wR}$$

The verification of the following result in Isabelle is then fully automatic.

Proposition 6 $B_1' \equiv B_1\ (\equiv B_2)$.

The B_2 axioms, in turn, are quite similar to the reflexive transitive closure axioms of relation algebra. In this setting, a relation x is reflexive if $1 \le x$, where regular algebra notation is used for the unit relation. Since binary relations form B_2 algebras, their axioms almost characterise x^* as the reflexive transitive closure of x. We have therefore added an explicit variant of B_2 as well.

Definition 6 A B_{rtc}-algebra is a star dioid which satisfies

$$1 + x + x^*x^* \le x^*, \tag{RTC1}$$

$$1 + x + yy \le y \Rightarrow x^* \le y. \tag{RTC2}$$

The verification of the following fact in Isabelle was again fully automatic.

Proposition 7 $B_{rtc} \equiv B_2\ (\equiv B_1 \equiv B_1')$.

6 Relative Completeness of Boffa's Axioms

At p.116 of his monograph, Conway conjectured completeness of his classical axioms expanded by a proposition called $P(G)$, where G denotes a finite semigroup or monoid. Written in suitable form, $P(G)$ can be associated with a system of equations over G. It has therefore called system of *monoid identities* by Krob. Conway's conjecture has been verified by Krob, but the particular techniques used in Krob's proof are not needed to understand the material of this article, and a formalisation in Isabelle is far beyond its scope. The monoid identities arise from considering matrix equations over regular algebras, and in particular the verification of the powerstar axiom in this setting (cf. the discussion at p.111-116 in Conway's monograph).

Boffa has called a C-algebra augmented by the rule $P(G)$ a C^+-algebra and used it in his relative completeness proof, which we have formalised in Isabelle.

Definition 7 Let G be a finite monoid. A C^+*-algebra* is a C-algebra expanded by the axiom

$$(\forall i, j \in G.\ x_i x_j \le x_{ij} \wedge x_{i,i}^* = x_{i,i}) \ \Rightarrow\ (\sum_{i \in G} x_i)^* = \sum_{i \in G} x_i, \tag{P(G)}$$

where $x_{i,j} = \sum_{ik=j} x_k$ and ij in x_{ij} denotes multiplication in G.

More precisely, in Conway's monograph, the second condition in the antecedent is $x_{i,j}^* = x_{i,j}$. However, Boffa has shown that his antecedent, which is an instance of Conway's, entails the consequent of the rule. Conway has shown that $P(G)$ is equivalent to a statement about matrices ([10] p.116). See [12,24] for further details.

Deriving (P(G)) in the context of B_1-algebras in Isabelle requires some preparation and auxiliary lemmas, of which we present the most important ones. We have programmed Boffa's argument directly. First, we have defined a polymorphic function type *boffa-mon* from finite (multiplicative) monoids into B_1-algebras to model elements x_i of B_1-algebras indexed by elements $i \in G$. We have also provided Isabelle syntax for $x_{i,j}$.

Lemma 5 *Let G be a finite monoid with identity* 1 *and B a B_1-algebra. Let $i \in G$ and $x_i, x_1, x_{1,1} \in B$. Then $\forall i \in G.\ x_{i,i}^* = x_{i,i}$ implies*

1. $x_1^* = x_1$,
2. $\sum_{i \in G} x_i = 1 + \sum_{i \in G} x_i$.

Proof

1. First, $x_{1,1} = \sum_{1k=1} x_k = x_1$, which can be verified in Isabelle by simplification. Therefore $x_1^* = x_{1,1}^* = x_{1,1} = x_1$, by simplification and automated theorem proving.
2. First, $x_1 = x_1^* = 1 + x_1^* = 1 + x_1$ holds by (1) and regular algebra. Thus

$$\sum_{i \in G} x_i = x_1 + \sum_{i \in G, i \neq 1} x_i = 1 + x_1 + \sum_{i \in G, i \neq 1} x_i = 1 + \sum_{i \in G} x_i.$$

□

Rearranging sums as in (2) generally requires some interaction with Isabelle.

Lemma 6 *Let G be a finite monoid and B a B_1-algebra. Let $i, j \in G$ and $x_i, x_j, x_{ij} \in B$. Then $\forall i, j \in G.\ x_i x_j \leq x_{ij}$ implies*

$$\sum_{i,j \in G} x_i x_j \leq \sum_{i \in G} x_i.$$

This may be trivial for humans to see, but its formalisation in Isabelle needed a number of auxiliary facts to rearrange and reason within sums.

The next statement formalises the main contribution of Boffa's first article.

Proposition 8 $B_1 \vdash C^+$.

Proof The axioms of C-algebras have been verified in Proposition 5, so it remains to derive $P(G)$. Suppose $\forall i, j \in G.\ x_i x_j \leq x_{ij}$ and $\forall i, j \in G.\ x_{i,i}^* = x_{i,i}$. Then, by Lemma 5(2) and Lemma 6,

$$(\sum_{i \in G} x_i)^2 = (1 + \sum_{i \in G} x_i)^2 = 1 + \sum_{i \in G} x_i + (\sum_{i \in G} x_i)^2 = \sum_{i \in G} x_i + \sum_{i,j \in G} x_i x_j = \sum_{i \in G} x_i.$$

Therefore, by (R) and again Lemma 5(2),

$$(\sum_{i \in G} x_i)^* = 1 + \sum_{i \in G} x_i = \sum_{i \in M} x_i.$$

□

This is essentially a direct formalisation of Boffa's proof in Isabelle; at the same level of granularity. From an automated reasoning point of view it is quite complex; our proofs combine reasoning about functions from finite monoids into regular algebras with calculations involving sums or suprema over finite sets. This is beyond first-order logic. Nevertheless, our proof document in the Archive shows that proof automation is reasonably high.

Since, by Krob's result, C^+ is complete for the equational theory of regular expressions, we obtain completeness of Boffa's algebras. This completeness result is relative because we have not attempted to formalise Krob's complicated proof in Isabelle.

Theorem 1 *Let $s, t \in \mathsf{Rex}(\Sigma)$. Then*

$$h(s) = h(t) \Rightarrow B_1 \vdash s = t \Leftrightarrow B_2 \vdash s = t \Leftrightarrow B_{rtc} \vdash s = t.$$

7 Conway's Conjectures

We now move one level down in Figure 1. Conway has presented several expansions of his classical axioms and conjectured their completeness (p.103). Boffa has verified one of them and Krob (p.329f) has considered the remaining ones relative to B_1. Krob seems to claim that all of Conway's quasi-identities lead to complete systems when added to the dioid axioms, (C11) and (C12) (p.330, Corollary 15.15). Our analysis with Isabelle yields a more fine grained view, showing that in some cases (C13) is needed as well.

Consider the following set of axioms.

$$xy = yz \;\Rightarrow\; x^*y = yz^*, \tag{C0}$$
$$xy \leq yz \;\Rightarrow\; x^*y \leq yz^*, \tag{C1l}$$
$$yx \leq zy \;\Rightarrow\; yx^* \leq z^*y, \tag{C1r}$$
$$x = yx \;\Rightarrow\; x = y^*x, \tag{C2l}$$
$$x = xy \;\Rightarrow\; x = xy^*, \tag{C2r}$$
$$xy \leq y \;\Rightarrow\; x^*y \leq y, \tag{C3l}$$
$$yx \leq y \;\Rightarrow\; yx^* \leq y. \tag{C3r}$$

Definition 8

- A C_0-algebra is a strong Conway dioid which satisfies (C0).
- A C_{1l}-algebra is a strong Conway dioid which satisfies (C1l).
- A C_{1r}-algebra is a strong Conway dioid which satisfies (C1r).
- A C_1-algebra is a C_{1l}-algebra and a C_{1r}-algebra.
- A C_{2l}-algebra is a Conway dioid which satisfies (C2l).
- A C_{2r}-algebra is a Conway dioid which satisfies (C2r).
- A C_2-algebra is a C_{2l}-algebra and a C_{2r}-algebra.
- A C_{3l}-algebra is a Conway dioid which satisfies (C3l).
- A C_{3r}-algebra is a Conway dioid which satisfies (C3r).
- A C_3-algebra is a C_{3l}-algebra and a C_{3r}-algebra.

Axiom (CX) has been called (P_X) by Conway; we follow Krob's notation. It is easy to see that the opposite of a C_{il}-algebra is a C_{ir}-algebra and vice versa. C_i-algebras are self-dual. As usual we have formalised these facts in Isabelle.

We first explain why strong Conway algebras are used in the first two definitions.

Lemma 7 *In C_0-algebras, C_{1l}-algebras, C_{1r}-algebras and C_1-algebras, axiom (C13) is independent.*

Proof The 3-element counterexample from Example 1 refutes $1^* = 1$ and $x^{**} = x^*$. □

Thus C_0-algebras, C_{1l}-algebras, C_{1r}-algebras and C_1-algebras without (C13) are incomplete.

The following proposition sums up the relationships between Boffa's algebras and Conway's conjectures, as shown in Figure 1.

Proposition 9

1. $C_{1l} \equiv C_{2l} \equiv C_{3l} \vdash B_1$,
2. $C_{1r} \equiv C_{2r} \equiv C_{3r} \vdash B_1$,
3. $C_0 \equiv C_1 \equiv C_2 \vdash C_{1l}$ *and* $C_0 \equiv C_1 \equiv C_2 \vdash C_{1r}$.

Proof The proof of (1) is by automated reasoning in Isabelle. (2) follows from opposition duality, which is picked up by Isabelle. (3) is immediate from (1) and (2). □

Completeness of Conway's conjectures then follows from that of Boffa's algebras.

Corollary 1 *Let* $s,t \in \mathsf{Rex}(\Sigma)$. *Then*

$$h(s) = h(t) \ \Rightarrow \ R \vdash s = t,$$

where R *is one of* C_0, C_{1l}, C_{1r}, C_1, C_{2l}, C_{2r}, C_2, C_{3l}, C_{3r}, C_3.

The question remains whether the entailment relations between Conway's and Boffa's algebras are strict. Section 9 provides a positive answer in terms of a counterexample.

8 Kleene Algebras

Kozen's Kleene algebras are the most widely studied and applied regular algebras. As for B_1' and B_{rtc}, the Kleene algebra axioms are particularly meaningful since the Kleene star is defined as a least pre-fixpoint and fixpoint. They have also turned out to be very useful in computing applications, which typically require reasoning under assumptions, and where Boffa's axioms seem to weak. In addition, Kleene algebras lend themselves for particularly simple completeness proofs which use Conway's trick of encoding finite automata in terms of matrix regular algebras over regular algebras. This proof has been formalised recently in Coq [8]. Boffa proved completeness of a simplified left-handed variant of Kleene algebra [7] relative to Krob's result. An alternative proof has recently been published by Kozen and Silva [23]. Our analysis sheds some new light on Kleene algebra, showing that they are, in fact, equipollent to Conway's variants.

Consider the following axioms.

$$1 + xx^* \leq x^*, \tag{Kl}$$
$$1 + x^*x \leq x^*, \tag{Kr}$$
$$xy \leq y \Rightarrow x^*y \leq y, \tag{C3l}$$
$$yx \leq y \Rightarrow yx^* \leq y, \tag{C3r}$$
$$z + xy \leq y \Rightarrow x^*z \leq y, \tag{C3l'}$$
$$z + yx \leq y \Rightarrow zx^* \leq y, \tag{C3r'}$$

where (C3l) and (C3r) have been used already in Section 7.

Definition 9

- A K_{1l}-algebra is a star dioid which satisfies (Kl) and (C3l).
- A K_{1r}-algebra is a star dioid which satisfies (Kr) and (C3r).
- A K_1-algebra is K_{1l}-algebra and a K_{1r}-algebra.
- A K_{2l}-algebra is a star dioid which satisfies (Kl) and (C3l').
- A K_{2r}-algebra is a star dioid which satisfies (Kr) and (C3r').
- A K_2-algebra is K_{2l}-algebra and a K_{2r}-algebra.

In that sense, Kleene algebras are simplified variants of Conway's C_3. It is well known that one of $1+xx^* \leq x^*$ and $1+x^*x \leq x^*$ is redundant in K_1 and K_2. Nevertheless, for reasons of symmetry and modular Isabelle development this is not of concern. We have formalised the obvious duality between left-handed and right-handed Kleene algebras in Isabelle.

In the axiomatisation of K_2, $1+x \cdot x^* \leq x^*$ implies $z+xx^*z \leq x^*z$. This formalises x^*z as the least pre-fixpoint (and least fixpoint) of the function $\lambda y.z+xy$. Similarly, x^* is the least pre-fixpoint of $\lambda y.1+xy$. A dual result holds for zx^*.

It is well known that K_1 and K_2 are equipollent. With Isabelle we have obtained a slightly more refined result fully automatically.

Lemma 8

1. $K_{1l} \equiv K_{2l}$.
2. $K_{1r} \equiv K_{2r}$.
3. $K_1 \equiv K_2$.

The next fact is more interesting.

Proposition 10

1. $K_{1l} \equiv C_{2l}$.
2. $K_{1r} \equiv C_{2r}$.
3. $K_1 \equiv C_2$.

The Isabelle proofs are again fully automatic. Proposition 10 immediately implies completeness of all variants of Kleene algebras.

Corollary 2 *Let* $s,t \in \mathsf{Rex}(\Sigma)$. *Then*

$$h(s) = h(t) \;\Rightarrow\; R \vdash s = t,$$

where R *is one of* K_{1l}, K_{1r}, K_1, K_{2l}, K_{2r}, K_2.

9 Separating the Algebras

We have now established equipollence, entailment and relative completeness results for Boffa's algebras, Conway's variants and Kozen's Kleene algebras, that is, the algebras in the central diamond in Figure 1. This section shows that the entailments in this diamond are strict. For this, it suffices to prove the following facts.

$$\begin{tikzcd}
 & K_{1l} \arrow[dr, phantom, sloped, "\nvdash"] & \\
B_1 \arrow[ur, phantom, sloped, "\nvdash"] \arrow[dr, phantom, sloped, "\nvdash"] & & K_1 \\
 & K_{1r} \arrow[ur, phantom, sloped, "\nvdash"] &
\end{tikzcd}$$

Kozen has provided a counterexample for $K_{1l} \vdash K_{1r}$ [21]. It shows that $K_{1l} \nvdash C3r$ and implies that $K_{1l} \nvdash K_1$. This counterexample is based on transfinite induction; its formalisation would have required the development of a substantial amount of background theory in Isabelle. The remaining relationships with Boffa's axiomatisations remain to be investigated. Fortunately Pratt gave a much simpler alternative [27] which requires only one step beyond the first limit ordinal. It is based on max-plus semirings, which have been formalised in Isabelle [1]. Pratt's counterexample can therefore be materialised with moderate effort. Since it is infinite, it is beyond the reach of Isabelle's counterexample generators. Extending Pratt's use of this counterexample, we show in particular strictness of $K_{1l} \vdash B_1$ and $K_{1r} \vdash B_1$.

Consider the set $\mathbb{N}_+$ which consists of $\mathbb{N}$ with the elements $\perp$, ∞ and $\top$ adjoined. The idea is that $\perp$ is below any natural number, whereas ∞ and $\top$ are above any natural number and $\infty < \top$. Informally speaking, therefore, this yields the linear order

$$\perp < 0 < 1 < 2 < \cdots < \infty < \top$$

which expands that on $\mathbb{N}$.

We have implemented $\mathbb{N}_+$ as an Isabelle datatype which expands $\mathbb{N}$. To obtain a max-plus semiring, we must interpret the maximum of two elements of $\mathbb{N}_+$ as semiring addition and addition in $\mathbb{N}_+$ as semiring multiplication. The element $\perp$ of $\mathbb{N}_+$ is interpreted as the semiring 0; the element 0 of $\mathbb{N}_+$ as the semiring 1. This requires expanding the functions max and $+$ from $\mathbb{N}$ to $\mathbb{N}_+$.

The definition of $\max : \mathbb{N}_+ \to \mathbb{N}_+ \to \mathbb{N}_+$ in Isabelle is straightforward. It is completely determined by the linear order on $\mathbb{N}_+$.

Addition on $\mathbb{N}$ is defined as usual. Since $\perp$ is the additive unit of the semiring and a multiplicative annihilator, $\perp + x = \perp$. Moreover, since 0 is the multiplicative unit, $\infty + 0 = \infty$ and $\top + 0 = \top$. More generally, $\top + x = \top$. This leaves choices for $\infty + \infty$, $x + \infty$ and $\infty + x$. In particular, addition does not need to be commutative beyond $\mathbb{N}$.

Pratt's insight has been that these choices can be used for forcing (Kl) and (Kr) and switching the axioms (C3l) and (C3r) alternatingly on and off. More precisely, $\infty + \infty = \top$ forces $1 + xx^* = x^*$ and $1 + x^*x = x^*$. Condition $1 + \infty = \top$ forces $xy \leq y \Rightarrow x^*y \leq y$, but not $yx \leq y \Rightarrow yx^* \leq y$. Condition $\infty + 1 = \top$ forces $yx \leq y \Rightarrow yx^* \leq y$, but not $xy \leq y \Rightarrow x^*y \leq y$.

Based on this insight one can define two different operations of addition on $\mathbb{N}_+$, which serve as multiplications on the corresponding semiring:

- $+_1$ where $m +_1 \infty = \infty$ for all $m \in \mathbb{N}$ and $\infty +_1 \alpha = \top$ for all $\alpha \in \mathbb{N}_+$;
- $+_2$ where $\infty +_2 m = \infty$ for all $m \in \mathbb{N}$ and $\alpha +_2 \infty = \top$ for all $\alpha \in \mathbb{N}_+$.

The construction so far is summed up in the following statement.

Proposition 11 *The following structures form* selective semirings*, that is, semirings in which $x + y = x$ or $x + y = y$ holds for all elements.*

1. $(\mathbb{N}_+, \max, +_1, \perp, 0)$,
2. $(\mathbb{N}_+, \max, +_2, \perp, 0)$.

The proofs in Isabelle are straightforward and automatic, but somewhat tedious due to case analyses on $\mathbb{N}_+$. All selective semirings are dioids (cf. [1]), hence $(\mathbb{N}_+, \max, +_1, \perp, 0)$ and $(\mathbb{N}_+, \max, +_2, \perp, 0)$ form dioids, as desired. Sledgehammer picks this up automatically.

It remains to consider the star. As a sum of powers, it is also determined by the order on $\mathbb{N}_+$. It maps $\perp$ and 0 to 0 and all other elements to $\top$. We can then define the structures

$$\mathcal{N}_1 = (\mathbb{N}_+, \max, +_1, \perp, 0, ^*), \qquad \mathcal{N}_2 = (\mathbb{N}_+, \max, +_2, \perp, 0, ^*)$$

and extend Proposition 11 to the following result, which has been verified in Isabelle.

Proposition 12

1. $\mathcal{N}_1 \models K_{1r}$*, but* $\mathcal{N}_1 \not\models C3l$*.*
2. $\mathcal{N}_2 \models K_{1l}$*, but* $\mathcal{N}_1 \not\models C3r$*.*

Proof

1. The verification of $1+xx^* \leq x^*$ and $xy \leq y \Rightarrow x^*y \leq y$ (to be translated into max-plus syntax) requires another case analysis over the elements of $\mathbb{N}_+$. To show that $(C3l)$ fails in $\mathscr{N}_1$ we have proved $\exists x,y.\neg(xy \leq y \wedge x^*y \leq y)$ in Isabelle. This is the case because

$$1 +_1 \infty = \infty < \top = \top +_1 \infty = 1^* +_1 \infty.$$

2. The proof is dual.

□

Proposition 12 yields the main theorem of this section.

Theorem 2

1. $B_1 \not\vdash K_{1l}$ and $B_1 \not\vdash K_{1r}$.
2. $K_{1l} \not\vdash K_1$ and $K_{1r} \not\vdash K_1$.

Proof $K_{1l} \vdash B_1$ and $K_{1r} \vdash B_1$ hold by Propositions 9 and 10. By Proposition 12, therefore, both $\mathscr{N}_1 \models B_1$ and $\mathscr{N}_2 \models B_1$. In addition, by Proposition 12, $\mathscr{N}_1 \not\models K_{1l}$ and $\mathscr{N}_2 \models K_{1l}$, which proves $B_1 \not\vdash K_{1l}$. Moreover, by Proposition 12, $\mathscr{N}_1 \models K_{1r}$ and $\mathscr{N}_2 \not\models K_{1r}$, which proves $B_1 \not\vdash K_{1r}$. Finally, again by Proposition 12, $\mathscr{N}_1 \not\models K_1$ and $\mathscr{N}_2 \not\models K_1$, which proves $K_{1l} \not\vdash K_1$ and $K_{1r} \not\vdash K_1$. □

Theorem 2 obviously adapts to equipollent algebras. In particular, it implies that $B_{rtc} \not\vdash K_1$, that is, the dioid axioms plus the standard reflexive transitive closure axioms are complete, but do not entail the Kleene algebra axioms. This answers a question from [17] (p.798).

10 Salomaa's Axioms

Salomaa [29] gave the first completeness proof for regular algebra and he conjectured completeness of another axiomatic variant (p.166). His axiomatisation is not based on universal identities or quasi-identities, but on axiom schemata ranging over regular expressions. This is because one of his inference rules is an algebraic abstraction of Arden's rule, which uses a syntactic side condition that encodes the absence of the empty word property. It expresses, at the level of regular expressions, the fact that a regular language does not contain the empty word; see [22] for further discussion.

Arden's rule is a well known tool for solving recursive equations over regular languages. Accordingly, Salomaa's completeness proof uses the corresponding axiom to show that each regular expression can be written as the solution of a recursive regular equation, and that for two equivalent regular expressions the corresponding coefficients in the associated regular equations must themselves denote the same regular language. Salomaa's axioms are based on dioids—more precisely dioid schemata—without 1. He defines 1 as 0^*, but it is obvious that dioids provide an equipollent basis in the presence of the star.

To circumvent the use of schematic inference rules and to align Salomaa's axioms with the other regular algebras studied in this article, we characterise the empty word property algebraically. This is justified for regular expressions in Proposition 14. Our characterisation is quite weak, but sufficient for proving completeness.

Consider the following axioms.

$$(1+x)^* = x^*, \tag{S11}$$

$$1+xx^* = x^*, \tag{S12l}$$

$$1+x^*x = x^*, \tag{S12r}$$

$$ewp(x) \;\Leftrightarrow\; \exists y.\, x = 1+y \;\wedge\; \neg ewp(y), \tag{EWP}$$

$$\neg ewp(x) \wedge y = z+xy \;\Rightarrow\; y = x^*z, \tag{Al}$$

$$\neg ewp(x) \wedge y = z+yx \;\Rightarrow\; y = zx^*. \tag{Ar}$$

Axioms (S11), (S12l) and (S12r) have already been verified in the context of Boffa's algebras (cf. Lemma 3). Axioms (S12l) and (S12r) are equational variants of Kozen's axioms (Kl) and (Kr). Axiom (EWP) is the weak characterisation of the empty word property mentioned. It expresses the fact that every language X which contains the empty word ε can be written as $X = \{\varepsilon\} \cup Y$ for some language Y which does not contain ε at the algebraic level. Axioms (Al) and (Ar) are algebraic abstractions of Arden's rule. For regular expressions $s,t \in \mathsf{Rex}(\Sigma)$, for instance, (Al) states that if an expression s does not have the empty word property, then the recursive equation $y = sy+t$ in y has the unique solution $y = s^*t$. At this level, the empty word property can be defined recursively (cf. Section 11).

More precisely, Arden's rules express *uniqueness* of the solutions of $y = sy+t$ in y and its opposition dual, whereas *existence* of these solutions follows from (S12l) and (S12r), that is, $ss^*t+t = s^*t$ and $ts^*s+t = ts^*$. It follows already in Boffa's algebras that

$$y = x^*z \;\Rightarrow\; y = z+xy, \qquad y = zx^* \;\Rightarrow\; y = z+yx,$$

hence (Al) and (Ar) can be strengthened to

$$\neg ewp(x) \;\Rightarrow\; (y = z+xy \Leftrightarrow y = x^*z), \qquad \neg ewp(x) \;\Rightarrow\; (y = z+yx \Leftrightarrow y = zx^*).$$

We can now loosely axiomatise Salomaa's algebra S_r, its dual S_l and a symmetric variant S which can be found at the bottom of the diagram in Figure 1.

Definition 10

- A S_l-algebra is a star dioid which satisfies (S11), (S12l), (EWP) and (Al).
- A S_r-algebra is a star dioid which satisfies (S11), (S12r), (EWP) and (Ar).
- A S-algebra is a S_l-algebra which is also a S_r-algebra.

It is easy to see that S_l and S_r are duals, whereas S is self-dual.

The question arises whether (S12l) and (S12r) can be weakened to (Kl) and (Kr). Nitpick rules this out, at least for our weak axiomatisation of *ewp*.

Example 2 Consider the structure defined by $0 < 1$, $0 < x$, $1 < y$, $x < y$, whereas x and 1 are incomparable. While this defines addition, the operations of multiplication and star are given by the tables

$\cdot$	0	1	x	y
0	0	0	0	0
1	0	1	x	y
x	0	x	y	y
y	0	y	y	y

	*
0	y
1	y
x	1
y	1

One can check that this structure forms a star dioid which satisfies (S11), (EWP), (Ar) and (Kr), but $0^* = y > 1+0 = 1+0\cdot 0^*$. Hence (S12r) fails and the resulting axiomatisation is incomplete. Notice that $0^* = 1$ fails in this example as well. □

A dual result holds for the left-handed version. The following fact is the key to completeness.

Lemma 9

1. $S_l \vdash K_{2l}$.
2. $S_r \vdash K_{2r}$.
3. $S \vdash K_2$.

Proof It suffices to verify (C3l') and (C3r'). We show that $S_r \vdash z + yx \leq y \Rightarrow zx^* \leq y$, which verifies (C3r'). The proof of $S_l \vdash C3l'$ then follows by duality. Suppose $z + yx \leq y$, that is, $z + yx + y = y$. We proceed by case analysis.

- If $\neg ewp(x)$, then $y = (y + z)x^*$ by (Ar) and therefore $zx^* \leq y$.
- If $ewp(x)$, then $x = 1 + w$ for some w with $\neg ewp(w)$ by (EWP). Then $y = z + yw + y$ and $zw^* \leq y$ by (Ar) as in the previous case and $zx^* \leq y$ by (S11).

□

We present this proof because it illustrates the typical style of reasoning with Salomaa's axioms: they require case analyses on *ewp* and using (EWP) to reduce the case where *ewp* holds to one where Arden's rule can be applied again. Supporting this style of reasoning is the main reason for introducing axiom (EWP).

Interestingly, $ewp(x)$ cannot be reduced to the condition $1 \leq x$, to which it corresponds in the language model.

Example 3 Consider the three-element structure where addition is defined by $0 < x < 1$, multiplication by $xx = x$ and the star by $0^* = x^* = 1^* = 1$. It can be checked that this structure forms a K_1-algebra where $1 \not\leq x$ and $0 + x \cdot 1 = x$, but $x < 1 \cdot 1 = 1 \cdot 1^*$. Thus, in K_1-algebras, $1 \not\leq x$ and $x = xy + z$ need not imply $x = zy^*$. □

Lemma 9 implies completeness of Salomaa's algebras.

Corollary 3 *Let $s, t \in \mathsf{Rex}(\Sigma)$. Then*

$$h(s) = h(t) \;\Rightarrow\; R \vdash s = t,$$

where R is one of S_l, S_r, S.

All the algebras in Figure 1 have now been proven complete relative to Krob's result. Lemma 9 is more fine-grained than Boffa's original one [6] which links S_r with B_1.

In addition, his proof contains a gap. He has shown that S_r entails a variant in which (Ar) is replaced by

$$\neg ewp(x) \wedge y = yx + 1 \Rightarrow y = x^*.$$

Verifying this entailment is trivial with Isabelle; its converse can be refuted with a three-element counterexample using Nitpick, though one should take this result with a grain of salt due to our loose axiomatisation of *ewp*. Salomaa has conjectured that this variant is complete as well. Boffa has argued that this variant, in turn, entails B_1. With Isabelle, we could neither verify nor refute his calculations. The problematic case in Boffa's proof is $(x + y)^* = (x^*y)^*x^*$, Boffa's axiom (C11), for which Boffa aims at showing with the above rule that $(x^*y)^*x^* = (x^*y)^*x^*(x + y) + 1$. Our attempts to verify or falsify this step with Isabelle were unsuccessful. In Boffa's article, a case analysis on *ewp* is missing. In particular for the cases where $ewp(x)$ or $ewp(y)$ holds we were neither able to find a proof nor a counterexample. In addition, direct proofs or refutations of (C11) with Isabelle failed. Given our experience and the power of Isabelle's automated theorem provers, doubts remain whether (C11) could be derived; a formal completeness proof for Salomaa's variant remains open.

11 Soundness of Salomaa's Axioms

So far we have compared the algebras in Figure 1 and verified their relative completeness. The regular algebra C^+ at the top of the diagram is complete by Krob's result and all other algebras entail C^+. In this section we prove soundness of all the algebras in the diagram by proving that the regular languages form a model of S, hence of all of its superclasses, which lie above it in Figure 1.

In fact it has already been shown with Isabelle [1] that languages and regular languages form dioids (Proposition 1) and Kleene algebras.

Proposition 13 *For each alphabet Σ, the following structures form Kleene algebras.*

1. $(\mathsf{Lan}(\Sigma), \cup, \cdot, \emptyset, \{\varepsilon\}, ^*)$,
2. $(\mathsf{Reg}(\Sigma), \cup, \cdot, \emptyset, \{\varepsilon\}, ^*)$.

In Isabelle, words over an alphabet are typically implemented as lists; the datatype of languages is that of sets of lists. This formalisation is clearly isomorphic to the usual language model. Regular expressions and the interpretation homomorphism $h : \mathsf{Rex}(\Sigma) \to \mathsf{Lan}(\Sigma)$ have also been implemented previously using Isabelle's **typedef** package. Our own development is built on these existing formalisations. A recent Isabelle implementation of regular expression equivalence [26] uses these formalisations, and so does a recent mechanisation of the Myhill-Nerode theorem in Isabelle [30].

By Proposition 13, all regular identities derived for Boffa's algebras in Isabelle can be used for proving that languages and regular languages form models of S-algebras. Relative to the Kleene algebra model, only the axioms (S11), (S12l), (EWP), (Al) and their duals need to be verified at the language level. However, (S11), (S12l) and (S12r) have already been verified in the context of B_2, hence they hold in the (regular) language model by Proposition 13 and because B_2 is a superclass of K_1 and K_2. Hence, in fact, only (EWP) and (Al) remain to be checked.

Axiom (EWP), that is $ewp(x) \Leftrightarrow \exists y.x = 1 + y \wedge \neg ewp(y)$, requires the instantiation of ewp at the language level. Obviously, for all $X \in \mathsf{Lan}(\Sigma)$, $ewp(X) \;\Leftrightarrow\; \varepsilon \in X$. Thus (EWP) holds at the language level by simple set-theoretic reasoning with $Y = X - \{\varepsilon\}$ as a witness for the existential quantifier.

Axiom (Al), that is $\neg ewp(x) \wedge y = xy + z \Rightarrow y = x^*z$, and its dual (Ar) require proving Arden's rule and its dual at the language level. This classical result has already been formalised in Isabelle [30]. Access to the algebraic layer makes our proof simpler and more abstract. The key algebraic properties are

$$y \leq yx + z \Rightarrow y \leq yx^{n+1} + zx^*$$

from Lemma 4(5) and its dual. They have been proved in the context of B_2-algebras and therefore hold in the language and regular language model. Only a few facts particular to the language model must be added. These describe the length increase of the shortest word in language YX^n proportional to n, whenever $Y \neq 0$ and $\varepsilon \notin X$.

Lemma 10 *Let $X, Y \in \mathsf{Lan}(\Sigma)$.*

1. *If $|v| \geq k$ for all $v \in X$, then $|w| \geq kn$ for all $w \in X^n$.*
2. *If $|u| \geq m$ and $|v| \geq n$ for all $u \in X$ and $v \in Y$, then $|w| \geq m + n$ for all $w \in XY$.*

We can then prove that no word in Y can be in YX^n for n sufficiently large.

Lemma 11 *Let $X, Y \in \mathsf{Lan}(\Sigma)$ with $\neg ewp(X)$ and $Y \neq \emptyset$. Then $\forall w \in Y.\exists n.w \notin YX^{n+1}$.*

We have also formalised the dual statement in Isabelle. Arden's rule (Ar) now follows from Lemma 4(5), Lemma 11 and basic set theory.

Lemma 12 *Let $X,Y,Z \in \mathsf{Lan}(\Sigma)$ with $\neg ewp(X)$. Then*

$$Y = YX \cup Z \Rightarrow Y = ZX^*.$$

Proof $Y = YX \cup Z$ implies $ZX^* \subseteq Y$ by axiom (C3r') of Kleene algebra, which holds in the language model by Proposition 13. So it remains to show that the assumption implies $Y \subseteq ZX^*$. We proceed by case analysis.

- If $Y = \emptyset$, then $Y \subseteq ZX^*$ holds in every dioid.
- Otherwise, Lemma 4(5) implies that every $w \in Y$ must also be in $YX^{n+1} \cup ZX^*$ for all $n \geq 0$, hence $w \in YX^{n+1}$ or $w \in ZX^*$ by set theory. Since the first alternative is ruled out by Lemma 11, this shows $Y \subseteq ZX^*$.

□

Again, we have verified (Al) in Isabelle, too, and we have established both variants of Arden's rule as equivalences. This has previously been mechanised at the language level in Isabelle [30]. With access to algebra our proofs are more automatic.

Arden's rule implies the following soundness theorem.

Theorem 3 *For every alphabet Σ, the structure $(\mathsf{Lan}(\Sigma), \cup, \cdot, \emptyset, \{\varepsilon\}, ^*)$ forms an S-algebra.*

Arden's rule fails of course in the (regular) language model without the constraint on the empty word property. We have materialised a counterexample in Isabelle.

Example 4 Consider the regular languages $X = Y = \{\varepsilon\}$ and $Z = \emptyset$. Then $YX \cup Z = Y = XY \cup Z$, but $X^*Z = ZX^* = Z \neq Y$. □

Finally we show the most important soundness result, namely that the *regular* languages form models of *S*-algebras. In Isabelle, regular languages have been defined as the images of regular expressions under the interpretation homomorphism. The type of regular languages has been defined as a subtype of the language type. Once more the axioms (S11), (S12l) and (S12r) hold because regular languages form Kleene algebras. The proofs of (Al) and (Ar) are straightforward as well from Lemma 12 and its dual, using type coercion between regular languages and languages. Hence it remains to verify axiom (EWP).

In the context of regular languages and regular expressions, it seems appropriate to use the well known definition of the empty word property on regular expressions by the inductive function $o : \mathsf{Rex}(\Sigma) \to \{0,1\}$ as

$$o(0) = 0, \qquad o(1) = 1, \qquad o(a) = 0,$$
$$o(s+t) = o(s) + o(t), \qquad o(st) = o(s) \cdot o(t), \qquad o(s^*) = 1.$$

It can be found in Salomaa's article. The correspondence between o and *ewp* is captured by the following lemma.

Lemma 13 *$o(s) = 1 \Leftrightarrow ewp(h(s))$ holds for all $s \in \mathsf{Rex}(\Sigma)$.*

We have also formalised the congruence $\sim$ from Section 2 in Isabelle and used it to prove the following analogue of axiom (EWP) in the algebra of regular expressions.

Proposition 14 *For all $s \in \mathsf{Rex}(\Sigma)$ there is a $t \in \mathsf{Rex}(\Sigma)$ such that $s \sim o(s) + t$ and $o(t) = 0$.*

The proof is by structural induction. Using this we have validated axiom (EWP) in the context of regular languages.

Lemma 14 *Let $X \subseteq \mathsf{Reg}(\Sigma)$. Then*

$$ewp(X) \Leftrightarrow \exists Y \in \mathsf{Reg}(\Sigma).\ X = \{\varepsilon\} \cup Y \wedge \neg ewp(Y).$$

Proof

$$\begin{aligned} ewp(X) &\Leftrightarrow \exists s \in \mathsf{Rex}(\Sigma).\ X = h(s) \wedge o(s) = 1 \\ &\Leftrightarrow \exists s,t.\ X = h(s) \wedge s \sim 1 + t \wedge o(t) = 0 \\ &\Leftrightarrow \exists t.\ X = \{\varepsilon\} \cup h(t) \wedge \neg ewp(h(t)) \\ &\Leftrightarrow \exists Y \in \mathsf{Reg}(\Sigma).\ X = \{\varepsilon\} \cup Y \wedge \neg ewp(Y), \end{aligned}$$

using Lemma 13 in the first and third step and Proposition 14 in the second one. □

This yields our ultimate soundness result for S.

Theorem 4 *For each alphabet Σ, the structure $(\mathsf{Reg}(\Sigma), \cup, \cdot, \emptyset, \{\varepsilon\}, {}^*)$ forms an S-algebra.*

It follows, in particular, that precise versions of Salomaa's axioms are valid in the regular language model. In addition, we have verified that regular expressions satisfy both variants of Arden's rule.

Lemma 15 *Let $x, y, z \in \mathsf{Rex}(\Sigma)$, Then*

$$o(x) = 0 \wedge y \sim z + xy \Rightarrow y \sim x^* z, \qquad o(x) = 0 \wedge y \sim z + yx \Rightarrow y \sim z x^*.$$

Similar results have been formalised previously in Isabelle [30], even the bi-implicational variants mentioned in Section 10. The connection with the algebraic layer, however, has not been considered in this work.

As usual, we obtain the following specific soundness result for identities.

Theorem 5 *Let $s, t \in \mathsf{Rex}(\Sigma)$. Then*

$$S \vdash s = t \ \Rightarrow\ h(s) = h(t).$$

The final result of this section collects all the individual soundness and completeness results for all the algebras in Figure 1. It follows from Theorem 1, Theorem 5 and our individual entailment results.

Theorem 6 *Let $s, t \in \mathsf{Rex}(\Sigma)$. Then*

$$R \vdash s = t \ \Leftrightarrow h(s) = h(t),$$

where R is one of C^+, B_1, B_2, B_{rtc}, C_{1l}, C_{2l}, C_{3l}, C_{1r}, C_{2r}, C_{3r}, C_0, C_1, C_2, C_3, K_{1l}, K_{2l}, K_{1r}, K_{2r}, K_1, K_2, S_l, S_r, S.

12 Other Variants

Many of the regular algebras considered so far have either been obtained from Conway dioids or from dioids satisfying unfold axioms like $1+xx^* = x^*$ and $1+x^*x = x^*$. In this section we swap the underlying bases and check completeness of the resulting algebras.

Section 8 shows that K_1-algebras and K_2-algebras with their left-handed and right-handed variants are equipollent. In addition, K_1 algebras can be seen as axiomatically simpler, but equipollent variants of C_3-algebras. Hence the following question arises: How do variants of C_3 algebras, in which the axioms (C3l) and (C3r), which are shared with K_1-algebras, are replaced by (C3l') and (C3r') from K_2-algebras, fit into the picture?

Definition 11

- A C'_{3l}-*algebra* is a Conway dioid which satisfies (C3l').
- A C'_{3r}-*algebra* a Conway dioid which satisfies (C3r').
- A C'_{3}-*algebra* a C'_{3l}-algebra which is also a C'_{3l}-algebra.

Lemma 16

1. $C'_{3l} \equiv C_{3l}$,
2. $C'_{3r} \equiv C_{3r}$,
3. $C'_{3} \equiv C_{3}$.

The Isabelle proof of these equipollence results were fully automatic. Lemma 16 thus adds new equipollent variants to Conway's conjectures and completes the picture with respect to Kozen's Kleene algebras.

Next we consider variants of Boffa's and Conway's algebras in which the star axioms for (strong) Conway dioids are replaced by the equational star unfold axioms (S12l) and (S12r), which are stronger equational variants of Kozen's axioms (Kl) and (Kr). We obtain the following incompleteness results.

Lemma 17

1. *There is a star dioid which satisfies (S12l), (S12r) and (R), but not (C11).*
2. *There is a star dioid which satisfies (S12l), (S12r) and (C0), but not* $1^* = 1$.
3. *There is a star dioid which satisfies (S12l), (S12r), (C1l) and (C1r), but not* $1^* = 1$.

Proof

1. Consider the five-element structure with addition defined by $0 < 1 < x < y < z$ and multiplication and star defined by the following tables.

$\cdot$	0	1	x	y	z
0	0	0	0	0	0
1	0	1	x	y	z
x	0	x	y	y	z
y	0	y	y	y	z
z	0	z	z	z	z

	$*$
0	1
1	1
x	z
y	y
z	z

It can be verified that this structure forms a star dioid which satisfies (S12l), (S12r) and (R), but $(x+y)^* = y^* = y < z = x^*(yx^*)^*$, which falsifies (C11).

2. The three-element structure defined in Example 1 forms a star dioid which satisfies (S12l), (S12r) and (C0), but not $1^* = 1$.

3. Again, the three-element structure defined in Example 1 can be used. It satisfies also (C1r) and (C1l).

□

The algebra in Lemma 17(1) is a variant of a B_1-algebra; that in (2) a variant of a C_0-algebra; that in (3) a variant of a C_1-algebra. Variants of C_3-algebras obtained in the same way imply the K_1-axioms. They are trivially sound and complete. Hence variants of B_2-algebras and C_2-algebras remain to be considered.

Let us first discuss a variant of a B_2 algebra obtained along the lines of Lemma 17, that is, a star dioid which satisfies (S12l), (S12r) and (B23). Pratt's two counterexamples hold in this algebra for obvious reasons, but the induction laws (C3l) and (C3r) can each be falsified by one of Pratt's algebras. In other words, this variant of B_2 is a strict superclass of left and right Kleene algebras. The simplest relative completeness proof would consist in deriving $1+x \leq x^*$ and $x^*x^* \leq x^*$ from these axioms. While verifying the first identity with Isabelle is easy, we could neither verify nor falsify the second one within Isabelle's default running times. Attempts to falsify a number of additional regular identities with Nitpick failed as well. We have not further attempted to derive the axioms of B_2-algebras or C^+-algebras and therefore leave completeness of this variant open.

Finally, we discuss a variant of C_2-algebras, which is a star dioid satisfying (S12l), (S12r), (C2l) and (C2r). For this variant, the obvious completeness proof would consist in deriving the star axioms (C11) and (C12) for Conway dioids. Once more we failed to verify or falsify these axioms within Isabelle's default running times. Alternatively we have unsuccessfully tried to derive (C3l) and (C3r) in order to obtain completeness relative to Kleene algebras and we have unsuccessfully tried to refute some additional regular identities. Completeness of this variant is therefore open as well.

Conceptually, our variant of B_2 is a mixture of a fixpoint-based and a reflexive-transitive-closure-based axiomatisation of the star, and therefore rather artificial. Verifying or refuting completeness of our variant of C_2 seems more appealing to further expand our understanding of the fine-structure of regular algebras.

13 Summary of Isabelle Proof Experience

Our entire mathematical development has been formalised in Isabelle/HOL. While some comments have been added to the text and more details can be found in the Appendix and the Isabelle proof document [13], this section briefly summarises our proof experience.

Our main aim was the elaboration of mathematical relationships and the provision of a reference formalisation. Proof automation was of secondary importance, and we have even aimed at step-wise proofs when these contained interesting details. For this reason we do not attempt a detailed quantitative analysis.

Our regular algebra libraries contain 211 facts—lemmas and subclass statements—in addition to a previous formalisation of dioids and Kleene algebras with nearly 700 facts. About 150 of the latter relate to dioids, finite suprema and (regular) languages. These are directly in the scope of our development. Within the regular algebra libraries, about 70% of proofs were obtained automatically by Sledgehammer while 30% are Isar proofs with user interaction. 433 of our proofs invoke *metis*; Isabelle's internal automated theorem prover which aims at reconstructing external proof outputs. Most of these were obtained automatically by Sledgehammer, sometimes after calling Isabelle's simplifier, or a tactic for induction or case analysis; about 20 of them needed some manual crafting. When Sledgehammer could not be used, we invoked Isabelle's simplifier 118 times, its internal automated deduction tools 59 times (*auto*, *blast*, *force*, etc.), and its induction tactic 22 times. Only 8 proofs employ a deductive micro-step with *rule_tac*. Automation had even been higher, $> 80\%$, with Isabelle's integrated SMT solvers. Using this techology, however, does not comply with current Archive of Formal Proofs requirements.

Even first-order equational reasoning in regular algebra is not entirely trivial for humans. Our high degree of automation may therefore seem surprising. One explanation is that libraries have been developed in a principled way based on mathematical experience and each Sledgehammer call had full access to all previously formalised facts.

We have generally experienced Isabelle as a mature well-balanced platform which offers a unique combination of expressivity for designing theory hierarchies and automated reasoning support for proofs and refutations. The formalisation of theory hierarchies and models with type classes was seamless apart from minor circularity problems in formalising opposition duality and some difficulties with type conversions when importing libraries. In the case of duality we have shown, for instance, that C_{1r}-algebras are duals of C_{1l}-algebra. We could prove the converse duality as well, but not import the associated dual theorems. This would lead to an infinite number of theorems being propagated into the theory name space which is impossible. In our formalisation, therefore, we have used duality and the associated theorem propagation only in one direction.

With a certain amount of mathematical knowledge, Isabelle's support for equational reasoning and counterexample search in first-order algebraic structures is certainly impressive, and an enhancement of mathematical practice. Beyond pure first-order reasoning, Isabelle's built-in provers and simplifiers usually allowed us to reconstruct proofs at least in a step-by-step fashion at textbook level, but their formalisation was sometimes tedious and the implementation of the necessary background theory could lead to a significant overhead.

Our experience highlights two main items on an imaginary Isabelle wish-list, which are certainly well known amongst Isabelle developers.

The first item is a more expressive type system and a more transparent framework for mathematical hierarchy design. One main issue is the importation of existing Isabelle libraries and the resulting type coercions, in particular those for finite sums and commutative monoids, which require non-trivial polymorphic constraints. Another one is the difficulty of reasoning with typed structures such as matrices which prevented us from formalising matrix-based algebras and attempting matrix-based completeness proofs in Isabelle.

The second item is the integration of external theorem provers à la Sledgehammer beyond pure first-order reasoning to complement Isabelle's built-in tools. One extension might be ACL2 [15] as support for inductive reasoning, another is LEO-II [2] which supports classical higher-order logic, and which is already supported by the current Isabelle version. Providing smoother transitions between first-order and higher-order reasoning could be crucial for making interactive theorem proving technology more attractive for mathematicians.

14 Conclusion

We have compared more than 20 variants of regular algebras from the literature and added a few new ones. We have verified entailment or subclass relationships between these algebras, mainly by automated theorem proving. We have also falsified these relationships, some of them via finite models obtained with Isabelle's counterexample generators, others by materialising an infinite model in Isabelle. We have proved completeness of all these variants relative to a complex result by Krob; we have verified soundness by showing with Isabelle that regular languages form models. We have also implemented a library for the weakest known finitary regular algebras, which is automatically available for all other ones. Many of the precise relationships between these algebras are new. Along the way we could simplify Boffa's axiomatisations, detect missing axioms in some of Krob's simplified versions of Conway's conjectures and find a gap in Boffa's proof for Salomaa's conjecture, which is not uncommon when formalising mathematics.

On the one hand, these results provide a fine-grained taxonomy of regular algebras. This may be of interest to those applying variants of Kleene algebras in computing applications. Many of these applications, however, require reasoning under assumptions, for which Boffa's algebras seem too weak and the case analysis required by Salomaa's axioms too unwieldy, and the use of models other than languages. Such considerations and applications are beyond the scope of this article.

On the other hand, our implementations are an interesting case study in formalised mathematics, for instance in automated theorem proving and counterexample search with interactive theorem provers—we would certainly not have attempted this particular investigation without this technology. Using Isabelle simplified the verification or falsification of entailment relations between algebraic axiom sets considerably; we could analyse large parts of Figure 1 literally in an afternoon. Our experience shows that, for axiomatic investigations of this kind, mechanised reasoning can be of great benefit to mathematicians. Higher-order aspects of formalised reasoning, however, for instance the derivation of the C^+-algebra axioms and Arden's rule, required tedious user interactions. Increasing the automation of shallow second-order proofs in Isabelle, for instance with sets and functions, seems particularly important for making this technology more appealing to mathematicians.

Our Isabelle development is accessible online from the Archive of Formal Proofs [13]. It links into an existing formalisation of variants of Kleene algebras [1], using mainly its classes for dioids and Kleene algebras as well as the (regular) language model, but the two hierarchies are generally orthogonal. The regular algebra hierarchy focuses on algebras which generate the same variety, but may have different quasi-varieties or elementary classes. The Kleene algebra hierarchy, in contrast, is built from generalisations of dioids to various near-semirings, where the star is axiomatised uniformly in the style of Kleene algebra. Those algebras are usually incomplete with respect to the equational theory of regular languages, yet interesting for computing applications. For many of these variants, completeness and decidability of the equational theory is open. Whether a combination of these two hierarchies leads to any interesting structures remains to be seen.

Acknowledgements We are grateful to Geoff Sutcliffe and the München Isabelle group for making ATP systems freely available over the Internet. We would like to thank Christian Urban and Tobias Nipkow for information on the implementation of regular languages in Isabelle, and the anonymous referees for comments and suggestions that helped us to improve the presentation of this article.

References

1. Armstrong, A., Struth, G., Weber, T.: Kleene algebra. Archive of Formal Proofs **2013** (2013)
2. Benzmüller, C., Sultana, N.: Leo-II Version 1.5. In: J.C. Blanchette, J. Urban (eds.) PxTP 2013, *EPiC Series*, vol. 14, pp. 2–10. EasyChair (2013)
3. Blanchette, J.C., Bulwahn, L., Nipkow, T.: Automatic proof and disproof in Isabelle/HOL. In: C. Tinelli, V. Sofronie-Stokkermans (eds.) FroCos 2011, *LNAI*, vol. 6989, pp. 12–27. Springer (2011)
4. Blanchette, J.C., Nipkow, T.: Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In: M. Kaufmann, L.C. Paulson (eds.) Interactive Theorem Proving, *LNCS*, vol. 6172, pp. 131–146. Springer (2010). DOI 10.1007/978-3-642-14052-5_11

5. Bloom, S.L., Ésik, Z.: Equational axioms for regular sets. Mathematical Structures in Computer Science **3**(1), 1–24 (1993)
6. Boffa, M.: Une remarque sur les systèmes complets d'identités rationnelles. Informatique théorique et applications **24**(4), 419–423 (1990)
7. Boffa, M.: Une condition impliquant toutes les identités rationnelles. Informatique théorique et applications **29**(6), 515–518 (1995)
8. Braibant, T., Pous, D.: An efficient Coq tactic for deciding Kleene algebras. In: M. Kaufmann, L. Paulson (eds.) ITP 2010, *LNCS*, vol. 6172, pp. 163–178. Springer (2010)
9. Bulwahn, L.: The new Quickcheck for Isabelle. In: C. Hawblitzel, D. Miller (eds.) Certified Programs and Proofs, *LNCS*, vol. 7679, pp. 92–108. Springer (2012)
10. Conway, J.H.: Regular Algebra and Finite Machines. Chapman and Hall (1971)
11. Eilenberg, S.: Automata, Languages and Machines. Academic Press (1976)
12. Ésik, Z.: Group axioms for iteration. Information and Computation **148**(2), 131–180 (1999)
13. Foster, S., Struth, G.: Regular algebras. Archive of Formal Proofs **2014** (2014)
14. Ginzburg, A.: Algebraic Theory of Automata. Academic Press (1968)
15. Gordon, M.J.C., Reynolds, J., Hunt, W.A., Kaufmann, M.: An integration of HOL and ACL2. In: FM-CAD 2006, pp. 153–160. IEEE Computer Society (2006)
16. Guttmann, W.: Algebras for iteration and infinite computations. Acta Informatica **49**(5), 343–359 (2012)
17. Höfner, P., Struth, G.: Algebraic notions of nontermination: Omega and divergence in idempotent semirings. J. Logic and Algebraic Programming **79**(8), 794–811 (2010)
18. Huffman, B., Kuncar, O.: Lifting and transfer: A modular design for quotients in Isabelle/HOL. In: G. Gonthier, M. Norrish (eds.) CPP 2013, *LNCS*, vol. 8307, pp. 131–146. Springer International Publishing (2013)
19. Iancu, M., Rabe, F.: Formalizing foundations of mathematics. Mathematical Structures in Computer Science **21**, 883–911 (2011)
20. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: C.E. Shannon, J. McCarthy (eds.) Automata Studies, chap. 1956, pp. 3–41. Princeton University Press (1956)
21. Kozen, D.: On Kleene algebras and closed semirings. In: B. Rovan (ed.) MFCS'90, *LNCS*, vol. 452, pp. 26–47. Springer (1990)
22. Kozen, D.: A completeness theorem for Kleene algebras and the algebra of regular events. Information and Computation **110**(2), 366–390 (1994)
23. Kozen, D., Silva, A.: Left-handed completeness. In: W. Kahl, T.G. Griffin (eds.) RAMiCS 2012, *LNCS*, vol. 7560, pp. 162–178. Springer (2012)
24. Krob, D.: Complete systems of $\mathscr{B}$-rational identities. Theoretical Computer Science **89**, 207–343 (1991)
25. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, *LNCS*, vol. 2283. Springer (2002)
26. Nipkow, T., Traytel, D.: Unified decision procedures for regular expression equivalence. In: G. Klein, R. Gamboa (eds.) ITP 2014, *LNCS*, vol. 8558, pp. 450–466. Springer (2014)
27. Pratt, V.: Action logic and pure induction. Tech. Rep. STAN-CS-90-1343, Department of Computer Science, Stanford University (1990)
28. Redko, V.N.: On the determining totality of an algebra of regular events. Ukrain. Math. Z. **16**, 120–126 (1964). (in Russian)
29. Salomaa, A.: Two complete axiom systems for the algebra of regular events. J. ACM **13**(1), 158–169 (1966)
30. Wu, C., Zhang, X., Urban, C.: A formalisation of the myhill-nerode theorem based on regular expressions. J. Autom. Reasoning **52**(4), 451–480 (2014)

Appendix A Axioms and Axiom Systems for Regular Algebra

A.1 Dioid axioms

$$(x+y)+z = x+(y+z) \qquad x+y = y+x \qquad x+0 = x \qquad x+x = x$$

$$(x\cdot y)\cdot z = x\cdot(y\cdot z) \qquad x\cdot 1 = x \qquad 1\cdot x = x$$

$$x\cdot(y+z) = x\cdot y+x\cdot z \qquad (x+y)\cdot z = x\cdot z+y\cdot z \qquad x\cdot 0 = 0 \qquad 0\cdot x = 0$$

A.2 Star axioms

$$(x+y)^* = (x^*\cdot y)^*\cdot x^* \tag{C11}$$

$$(x\cdot y)^* = 1+x(y\cdot x)^*\cdot y \tag{C12}$$

$$x^{**} = x^* \tag{C13}$$

$$x^* = (x^{n+1})^*\cdot x_0^n \tag{C14n}$$

$$x\cdot x = x \;\Rightarrow\; x^* = 1+x \tag{R}$$

$$1+x \leq x^* \tag{B21}$$

$$x^*\cdot x^* = x^* \tag{B22}$$

$$1+x \leq y \wedge y\cdot y = y \;\Rightarrow\; x^* \leq y \tag{B23}$$

$$x\cdot x \leq x \;\Rightarrow\; x^* = 1+x \tag{wR}$$

$$1+x+x^*\cdot x^* \leq x^* \tag{RTC1}$$

$$1+x+y\cdot y \leq y \;\Rightarrow\; x^* \leq y \tag{RTC2}$$

$$x\cdot y = y\cdot z \;\Rightarrow\; x^*\cdot y = y\cdot z^* \tag{C0}$$

$$x\cdot y \leq y\cdot z \;\Rightarrow\; x^*\cdot y \leq y\cdot z^* \tag{C1l}$$

$$y\cdot x \leq z\cdot y \;\Rightarrow\; y\cdot x^* \leq z^*\cdot y \tag{C1r}$$

$$x = y\cdot x \;\Rightarrow\; x = y^*\cdot x \tag{C2l}$$

$$x = x\cdot y \;\Rightarrow\; x = x\cdot y^* \tag{C2r}$$

$$x\cdot y \leq y \;\Rightarrow\; x^*\cdot y \leq y \tag{C3l}$$

$$y\cdot x \leq y \;\Rightarrow\; y\cdot x^* \leq y \tag{C3r}$$

$$1+x\cdot x^* \leq x^* \tag{Kl}$$

$$1+x^*\cdot x \leq x^* \tag{Kr}$$

$$z+x\cdot y \leq y \;\Rightarrow\; x^*\cdot z \leq y \tag{C3l'}$$

$$z+y\cdot x \leq y \;\Rightarrow\; z\cdot x^* \leq y \tag{C3r'}$$

$$(1+x)^* = x^* \tag{S11}$$

$$1+x\cdot x^* = x^* \tag{S12l}$$

$$1+x^*\cdot x = x^* \tag{S12r}$$

$$ewp(x) \;\Leftrightarrow\; \exists y.\, x = 1+y \;\wedge\; \neg ewp(y) \tag{EWP}$$

$$\neg ewp(x) \wedge y = z+x\cdot y \;\Rightarrow\; y = x^*\cdot z \tag{Al}$$

$$\neg ewp(x) \wedge y = z+y\cdot x \;\Rightarrow\; y = z\cdot x^* \tag{Ar}$$

A.3 Axiom Systems

- Conway dioid: dioid axioms + (C11) + (C12)
- strong Conway dioid: Conway dioid + (C13)
- C-algebra: strong Conway dioid + (C14n)
- B_1-algebra: Conway dioid + (R)
- B_2-algebra: dioid axioms + (B21) + (B22) + (B23)
- B'_1-algebra: Conway dioid + (wR)
- B_{rtc}-algebra: dioid axioms + (RTC1) + (RTC2)
- C_0-algebra: strong Conway dioid + (C0)
- C_{1l}-algebra: strong Conway dioid + (C1l)
- C_{1r}-algebra: strong Conway dioid + (C1r)
- C_1-algebra: C_{1l}-algebra + C_{1r}-algebra
- C_{2l}-algebra: Conway dioid + (C2l)
- C_{2r}-algebra: Conway dioid + (C2r)
- C_2-algebra: C_{2l}-algebra + C_{2r}-algebra
- C_{3l}-algebra: Conway dioid + (C3l)
- C_{3r}-algebra: Conway dioid + (C3r)
- C_3-algebra: C_{3l}-algebra + C_{3r}-algebra
- K_{1l}-algebra: dioid axioms + (Kl) + (C3l)
- K_{1r}-algebra: dioid axioms + (Kr) + (C3r)
- K_1-algebra: K_{1l}-algebra + K_{1r}-algebra
- K_{2l}-algebra: dioid axioms + (Kl) + (C3l')
- K_{2r}-algebra: dioid axioms + (Kr) + (C3r')
- K_2-algebra: K_{2l}-algebra + K_{2r}-algebra
- S_l-algebra: dioid axioms + (S11) + (S12l) + (EWP) + (Al)
- S_r-algebra: dioid axioms + (S11) + (S12r) + (EWP) + (Ar)
- S-algebra: S_l-algebra + S_r-algebra

Appendix B Behind the scenes

This appendix provides additional information on our formalisation of regular algebras in Isabelle/HOL in the Archive of Formal Proofs [13]: on type classes, the **typedef** package, the lifting package, the Isar proof scripting language, the Sledgehammer tool for integrating external first-order theorem provers and the counterexample generator Nitpick. Our intention is to support the interested reader in consulting the Archive proof document alongside this article. Isabelle's excellent documentation contains more detailed background information.

B.1 Type Classes

Isabelle's type classes support the modular incremental specification of algebraic hierarchies. A type class consists of a signature declaration and a collection of axioms over this signature. Each type class gives rise to a local theory context in which the signature functions and axioms are in scope for deriving further facts. These contexts are local in that the axioms do not augment Isabelle's own axiom space, but exist only with respect to possible models. Type classes without models cannot be instantiated. Instantiations provide a concrete model type and concrete signature functions for this type together with proofs that the given assumptions hold for the functions instantiated. Instantiations export all derived laws for a particular type class into the global theory context for the given model type.

The following type classes axiomatise Conway dioids and strong Conway dioids:

class conway-dioid = star-dioid +
  assumes C11: $(x + y)^\star = (x^\star \cdot y)^\star \cdot x^\star$
  and C12: $(x \cdot y)^\star = 1 + x \cdot (y \cdot x)^\star \cdot y$

class strong-conway-dioid = conway-dioid +
  assumes C13: $(x^\star)^\star = x^\star$

In this example, the class of Conway dioids expands that of star dioids by adding the axioms (C11) and (C12). Moreover, a strong Conway dioid is a Conway dioid expanded by (C13). The entire regular algebra hierarchy has been defined this way by expanding type classes. For example, we can further expand to Conway algebras.

class C-algebra = strong-conway-dioid +
  assumes C14: $x^\star = (x^{n+1})^\star \cdot x_0{}^n$

B.2 Isar Proofs

Within a type class context a variety of facts are in scope for deriving further ones. This includes the axioms provided, all facts proved in the context of super-classes, and all facts which have already been verified in the current context. As an example we prove the dual version of (C11) in the context of Conway dioids.

lemma (**in** conway-dioid) C11-var: $(x + y)^\star = x^\star \cdot (y \cdot x^\star)^\star$
proof –
  have $x^\star \cdot (y \cdot x^\star)^\star = x^\star + x^\star \cdot y \cdot (x^\star \cdot y)^\star \cdot x^\star$
    by (metis C12 distrib-left mult-assoc mult-oner)
  also have ... $= (1 + x^\star \cdot y \cdot (x^\star \cdot y)^\star) \cdot x^\star$
    by (metis distrib-right mult-assoc mult-onel)
  finally show ?thesis
    by (metis C11 C12 mult-onel mult-oner)
qed

The proof, which uses Isabelle's proof scripting language Isar, is by and large human readable. It uses axioms such as (C11) and (C12) and facts such as *distrib-left* or *mult-assoc*, which have previously been verified in the context of a Conway dioid, and which are listed in the Isar proof. Essentially a "**have**" line proves an intermediate fact, "**also**" transitively chains a previous proof step to the next one, and "**show**" discharges a top-level proof goal. Statements of the form "**by** (*metis* ...)" are explained in the next section of this appendix.

B.3 Sledgehammer

In the above example, we have used Isabelle's Sledgehammer tool to discharge each individual proof goal. It calls a number of external first-order automated theorem provers on a given proof goal. These provers are provided with a set of assumptions available in a class context together with the goal. They are executed within a given time limit. If a proof is found, the assumptions used and the goal are handed over to the internal theorem prover *Metis*, which aims to reconstruct a proof with respect to Isabelle's axioms. The results, in particular the lists of assumptions used, are documented in statements of the form "**by** (*metis* ...)" in the above example proof. Internal proof reconstruction ensures that Isabelle developments do not rely on the soundness of the external tools, which are highly complex, but entirely on Metis which has been verified within Isabelle/HOL and thus builds on its small trustworthy kernel (see [3] for more details). In practice, Metis is much less effective than the external theorem provers and often not able to reconstruct their proofs.

B.4 Nitpick and Quickcheck

Aside from Sledgehammer, we have used Isabelle's Nitpick tool to search for finite counterexamples, as the following example illustrates.

lemma (**in** conway-dioid) $1^\star = 1$
  nitpick

For this particular conjecture, Nitpick found a counterexample with 3 elements and provided definitions for all relevant operations of the underlying type class (omitted for brevity).

Nitpick found a counterexample for card $'a = 3$:
Free variables:
op $<$ =
$\lambda x.\_$
$(a_1 := (\lambda x.\_)(a_1 := \mathit{False}, a_2 := \mathit{False}, a_3 := \mathit{False}),$
$(a_2 := (\lambda x.\_)(a_1 := \mathit{True}, a_2 := \mathit{False}, a_3 := \mathit{True}),$
$(a_3 := (\lambda x.\_)(a_1 := \mathit{True}, a_2 := \mathit{False}, a_3 := \mathit{False}))$
$0 = a_2$, $1 = a_3$, op $\leq$ = $\cdots$, op $+$ = $\cdots$, op $\cdot$ = $\cdots$, star = $\cdots$

Nitpick generates a type consisting of three elements, $a_1 \cdots a_3$ together with tabular definitions for each operation. The second line for operation $<$, for instance, states that $a_2 < a_1$, $a_2 \not< a_2$ and $a_2 < a_3$. Isabelle's counterexample generator Quickcheck works similarly. Several of our counterexamples could be obtained this way. A notable exception is Pratt's infinite counterexample from Section 9, which separates the algebras in Figure 1, and which had to be materialised explicitly in Isabelle. Counterexamples presented by Nitpick are not verified by Isabelle by default. It is, however, possible to invoke Nitpick with a flag that allows their verification by `auto`. In the above case this fails because of the anonymous terms of the form $(\lambda x.\_)$. The counterexamples in this article need therefore to be checked manually or trusted. In fact, Nitpick is based on a SAT-solver that is widely used in formal methods and program verification.

B.5 Typedefs and Lifting

Types can be specified in Isabelle/HOL by identifying a suitable subset of an existing type. As an example we show how the type of regular languages, for is created from the homomorphic image of $\mathsf{Reg}(\Sigma)$ under the interpretation homomorphism h, that is, from a subset of the type of languages. This example is taken from [1], but similar Isabelle formalisations of languages and regular languages have been given before. A language *'a lan* in Isabelle is simply represented as a set of polymorphic lists, that is, as a synonym of type *'a list set*. The datatype *'a rexp* of regular expressions provides the syntax of polymorphic regular expressions over arbitrary alphabets. The interpretation homomorphism h is called *lang* in Isabelle. It has been used for generating the regular languages precisely as described in Section 2. Hence the type of regular languages can be defined directly and succinctly as the homomorphic image of *lang*:

```
typedef 'a reg-lan = range lang :: 'a lan set
 by auto
```

Types in Isabelle/HOL must be non-empty to ensure consistency. Their definitions must therefore be justified by inhabitance proofs. In the above example, a trivial call of Isabelle's internal *auto* tactic sufficed, since the image of a function must produce at least one value.

Type definitions of the above kind come with two coercion functions: The first one is an abstraction function—a partial injection of values of the existing type into the new type—in this case *Abs-reg-lang* :: *'a lan* ⇒ *'a reg-lan*. The second one is a representation function which injects values of the new type into the old one. In this particular case it has type *Rep-reg-lang* :: *'a reg-lan* ⇒ *'a lan*.

Isabelle supports the definition of functions over types defined by **typedef** by lifting from the underlying type. This is supported by the **lifting** package [18], which requires a proof that the underlying function is closed over the type's characteristic set. For example language union can be lifted to regular languages since union preserves regularity.

```
lift-definition plus-rlan :: 'a reg-lan ⇒ 'a reg-lan ⇒ 'a reg-lan
is plus by (metis (hide-lams, no-types) image-iff lang.simps(4) rangeI)
```

A specific type for this function is obtained by relating it to the underlying function *op* + on languages—which *is* plus—and providing a closure proof, here by Sledgehammer and Metis. All regular operations have been formalised like this. Properties of functions on languages then lift automatically to regular expressions. As an example we show the proof that regular languages over a given alphabet form a dioid via a type class instantiation.

```
instantiation reg-lan :: (type) dioid
begin

 lift-definition zero-reg-lan :: 'a reg-lan is 0 by (metis lang.simps(1) rangeI)

 lift-definition one-reg-lan :: 'a reg-lan is 1 by (metis lang.simps(2) rangeI)

 lift-definition less-eq-reg-lan :: 'a reg-lan ⇒ 'a reg-lan ⇒ bool is less-eq .

 lift-definition less-reg-lan :: 'a reg-lan ⇒ 'a reg-lan ⇒ bool is less .
 lift-definition plus-reg-lan :: 'a reg-lan ⇒ 'a reg-lan ⇒ 'a reg-lan
  is plus by (metis (hide-lams, no-types) image-iff lang.simps(4) rangeI)

 lift-definition times-reg-lan :: 'a reg-lan ⇒ 'a reg-lan ⇒ 'a reg-lan
  is times by (metis (hide-lams, no-types) image-iff lang.simps(5) rangeI)

 instance proof
  fix x y z :: 'a reg-lan
  show x + y + z = x + (y + z)
   by transfer (metis join-semilattice-class.add-assoc')
  show x + y = y + x
   by transfer (metis join-semilattice-class.add-comm)
  show x · y · z = x · (y · z)
   by transfer (metis semigroup-mult-class.mult.assoc)
  show (x + y) · z = x · z + y · z
   by transfer (metis semiring-class.distrib-right)
  show x ≤ y ⟷ x + y = y
   by transfer (metis plus-ord-class.less-eq-def)
  show x < y ⟷ x ≤ y ∧ x ≠ y
   by transfer (metis plus-ord-class.less-def)
  show x + x = x
   by transfer (metis join-semilattice-class.add-idem)
  show x · (y + z) = x · y + x · z
   by transfer (metis semiring-class.distrib-left)
 qed
end
```

Each regular language operation has been defined by lifting the corresponding language operation. The dioid laws have then then been proved in the **instance** proof. Each proof proceeds by first applying **transfer**, which maps the proof to the underlying language type, and then calling Sledgehammer. This approach allowed us to verify that regular languages form a model of most of the regular algebras in this article. Only Salomaa's algebras required a special treatment, which is described in Section 11.